Available online at www.sciencedirect.com

ScienceDirect

Procedia Economics and Finance 10 (2014) 68 – 77

Procedia Economics and Finance

www.elsevier.com/locate/procedia

7th International Conference on Applied Statistics

A SAS approach for estimating the parameters of an alpha-stable distribution

Daniel Traian Pele[a] 1

[a] The Bucharest University of Economic Studies, 6 Piata Romana, Bucharest 010374, Romania

Abstract

Although there are several software products dealing with the issue of simulating and estimating a stable distribution, SAS has no procedure for stable distributions. In this paper we propose two macros for estimating the parameters of a stable distribution using McCulloch method and Kogon-Williams method; further developments are required for implementing a procedure for estimating the parameters of a stable distribution using maximum likelihood method.

Selection and peer-review under responsibility of the Department of Statistics and Econometrics, Bucharest University of Economic Studies.

Keywords: stable distributions; SAS macro; heavy-tails

1. Introduction

An alternative to the Gaussian distribution is the use of heavy-tailed distributions whose probability associated to tails is greater than in the case of the normal distribution, which allows for more robust estimates for the probability of occurrence of extreme events.

One of the researchers who influenced decisively the modelling of financial phenomena, Benoit Mandelbrot, proposed (1963), the so called stable distribution (Pareto-Levy), for the behaviour of stock market returns, that explains better than normal distribution the likelihood of extreme values.

Although in terms of the mathematical formalism the stable distributions have been studied extensively over the past decades, their application in financial modelling occurred relatively late, since the 1990s, as the practical use of this mathematical tool requires a large computational effort.

Stable distributions are a class of distributions having the property of being invariant under linear combinations; Gaussian distribution is a special case of stable distributions. There are numerous studies regarding the application of stable distributions in modelling financial phenomena, recalling here the works of Rachev and J.P. Nolan who brought important contributions to the use of stable distributions in finance.

Thus, Rachev (2003) and Rachev and Mittnik (2000) describe in their papers, in detail, the methods of estimation for such distributions.

* Corresponding author.
E-mail address: danpele@ase.ro.

Selection and peer-review under responsibility of the Department of Statistics and Econometrics, Bucharest University of Economic Studies.
doi:10.1016/S2212-5671(14)00279-2

Nolan (2011) proposes an efficient method for estimating parameters of stable distributions, using the maximum likelihood method with numerical approximations for the probability density function; his program, *stable.exe*, is a tool very used in practical applications.

In terms of the software tools used for simulation and estimation of stable distributions, several software applications can be mentioned:

- stable. exe (Nolan, 2005), with versions for R, Mathematica, Matlab and Excel;
- XploRe (Härdle, Klinke, Turlach, 1995), a software handling estimation of stable distributions using various methods;
- Various functions developed for Matlab, like Stableregkw (Weron, 2010), or R, like fBasics package.

Usually, SAS software is deeply involved in basic manipulation of large datasets extracted from financial platforms, either daily or intraday data regarding stocks, bonds, options. This only a primary step, as for complex estimations, SAS datasets are converted in formats readable for Matlab or R, since for best our knowledge, there is no SAS macro or SAS procedure for dealing with simulation and estimation of stable distributions.

In this paper we introduce a macro developed under SAS 9.3 for estimating the parameters of an alpha-stable distribution, using the time-series regression method from Kogon and Williams (1998).

The paper is organized as follows: a brief introduction in the formalism of stable distribution, including some simulation and estimation methods and the SAS macros for estimating the parameters of an alpha-stable distribution.

2. Stable distributions – simulation and estimation

2.1. Stable distributions

The difficulty that occurs for stable distributions is that in most situations an explicit form of the probability density function does not exist, but only the expression of the characteristic function is known.

Thus, a random variable X follows a stable distribution with parameters $(\alpha,\beta,\gamma,\delta)$ (Nolan, 2011) if exists $\gamma > 0, \delta \in \mathbf{R}$ such as X and $\gamma Z + \delta$ have the same distribution, where Z is a random variable with the characteristic function

$$\phi(t) = \mathbf{E}[e^{itZ}] = \begin{cases} \exp(-|t|^{\alpha}[1 - i\beta \tan(\frac{\pi\alpha}{2}) sign(t)]), \alpha \neq 1 \\ \exp(-|t|[1 + i\beta t \frac{2}{\pi} sign(t)(\ln(|t|)]), \alpha = 1 \end{cases}. \tag{1}$$

In the above notations $\alpha \in (0.2]$ is the stability index, controlling for probability in the tails (for Gaussian distribution $\alpha = 2$), $\beta \in [-1,1]$ is the skewness parameter, $\gamma \in (0,\infty)$ is the scale parameter and $\delta \in \mathbf{R}$ is the location parameter.

A random variable X follows a stable distribution $S(\alpha,\beta,\gamma,\delta;0)$ if his characteristic function has the form

$$\phi(t) = \mathbf{E}[e^{itX}] = \begin{cases} \exp(-\gamma^{\alpha}|t|^{\alpha}[1 + i\beta \tan(\frac{\pi\alpha}{2}) sign(t)(|\gamma t|^{1-\alpha} - 1)] + i\delta t), \alpha \neq 1 \\ \exp(-\gamma|t|[1 + i\beta t \frac{2}{\pi} sign(t)(\ln(|\gamma t|)] + i\delta t), \alpha = 1 \end{cases}. \tag{2}$$

A random variable X follows a stable distribution $S(\alpha,\beta,\gamma,\delta;1)$ if his characteristic function has the form

$$\phi(t)=\mathbf{E}[e^{itX}]=\begin{cases}\exp(-\gamma^{\alpha}|t|^{\alpha}[1-i\beta\tan(\frac{\pi\alpha}{2})sign(t)]+i\delta t),\alpha\neq 1\\ \exp(-\gamma|t|[1+i\beta t\frac{2}{\pi}sign(t)(\ln(|t|)]+i\delta t),\alpha=1\end{cases}. \quad (3)$$

The parameterisation $S(\alpha,\beta,\gamma,\delta;1)$ has the advantage that is more suitable for algebraic manipulations, although his characteristic function is not continuous for all parameters.

The parameterisation $S(\alpha,\beta,\gamma,\delta;0)$ is suitable for numerical simulations and statistical inference, although the expression of characteristic function is more difficult to utilise in algebraic calculus.

Nolan(2011) shows that the two parameterisations are equivalent; if $X \sim S(\alpha,\beta,\gamma,\delta_1;1)$ and $X \sim S(\alpha,\beta,\gamma,\delta_0;0)$, then $\delta_0=\begin{cases}\delta_1+\beta\gamma\tan\frac{\pi\alpha}{2},\alpha\neq 1\\ \delta_1+\beta\frac{2}{\pi}\gamma\ln\gamma,\alpha=1\end{cases}$. (4)

The behavior of stable distributions is driven by the values of stability index α : small values are associated to higher probabilities in the tails of the distribution.

2.2. Simulating a stable distribution

For simulating an alpha-stable distribution, under the parameterization S1, the following algorithm is widely used (Weron, 1996):

- Generate a uniformly distributed random variable $U \sim Unif\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$ and an exponential random variable $E \sim Exp(1)$;
- If $\alpha\neq 1$ then let $B(\alpha,\beta)=\frac{\arctan\left(\beta\tan\frac{\pi\alpha}{2}\right)}{\alpha}$, let $S(\alpha,\beta)=\left(1+\beta^2\tan^2\frac{\pi\alpha}{2}\right)^2$ and define the random variable $X=S(\alpha,\beta)\frac{\sin(U+B(\alpha,\beta))}{(\cos U)^{1/\alpha}}\left(\frac{U-\alpha(U+B(\alpha,\beta))}{E}\right)^{\frac{1-\alpha}{\alpha}}$;
- If $\alpha=1$ then define the random variable $X=\frac{2}{\pi}\left(\frac{\pi}{2}+\beta U\right)\tan U-\beta\ln\frac{\frac{\pi}{2}E\cos U}{\frac{\pi}{2}+\beta U}$;
- Finally, the random variable $Y=\begin{cases}\gamma X+\delta,\ \alpha\neq 1\\ \gamma X+\frac{2}{\pi}\beta\gamma\ln\gamma+\delta,\ \alpha=1\end{cases}$ follows a stable distribution $S(\alpha,\beta,\gamma,\delta;1)$.

2.3. Estimating parameters of a stable distribution using McCulloch method

McCulloch (1986) method involves the following steps for estimating the parameters of a $S(\alpha,\beta,\gamma,\delta;0)$ random variable:

- estimate α and β, using the quintiles of the empirical distribution (for more details, see Racheva-Iotova (2010);

- define $v_\alpha = \frac{x_{0.95} - x_{0.05}}{x_{0.75} - x_{0.25}}$ and $v_\beta = \frac{x_{0.95} + x_{0.05} - 2x_{0.25}}{x_{0.95} - x_{0.05}}$, where x_p is the p quintile of the empirical distribution, having thus $v_\alpha = \phi_1(\alpha,\beta)$ and $v_\beta = \phi_2(\alpha,\beta)$ or, by inversion, $\alpha = \psi_1(v_a, v_\beta)$ and $\beta = \psi_2(v_a, v_\beta)$.

More, $\alpha = \psi_1(v_a, v_\beta) = \psi_1(v_a, -v_\beta)$ and $\beta = \psi_2(v_a, v_\beta) = -\psi_2(v_a, -v_\beta)$.

The functions $\psi_1(\cdot)$ and $\psi_2(\cdot)$ are tabulated for different values of v_a and v_b, so the estimates of α and β can be obtained using a bi-linear interpolation.

In a quite similar manner, the location parameter δ and the scale parameter γ can be estimated using the corresponding tabulated functions and the previous estimations for α and β.

For the implementation of this method, a SAS macro called *%mc_culloch(data=,var=)* (see Appendix B) have been developed, with two arguments: *data* – the SAS dataset and *var* – the random variable for which we estimate the parameters of a stable distribution; the estimated values are saved in a SAS dataset called *parms*.

2.4. Estimating parameters of a stable distribution using Kogon-Williams method

In order to estimate the parameters of a stable distribution in parameterisation S1, the following algorithm can be applied (Kogon, Williams, 1998):

Step 1. Define the maximum error for convergence *error* and the maximum number of iterations *maxiter*;

Step 2. Use the initial estimates $\alpha_0, \beta_0, \gamma_0, \delta_0$ from McCulloch method and normalize the sample: $x_j \to \frac{x_j - \delta_0}{\gamma_0}$;

Step 3. Estimate the regression model $y_k = b + \alpha_1 w_k + \varepsilon_k$, with $k = 0,..,9$, $y_k = \ln[-\mathrm{Re}[\ln(\hat{\phi}(u_k))]]$, $w_k = \ln|u_k|$, $u_k = 0.1 + 0.1k$, $k = 0,..,9$, and $\hat{\phi}(\cdot)$ is the empirical characteristic function of the normalized sample. If $\hat{b}$ and $\hat{\alpha}_1$ are the estimates of the regression model, then the estimate of the scale parameter is $\hat{\gamma}_1 = \exp(\hat{b} / \hat{\alpha})$.

Step 4. Estimate the regression model $z_k = \delta_{11} + \beta_1 v_k + \eta_k$, with $k = 0,..,9$, $z_k = \mathrm{Im}[\ln(\hat{\phi}(u_k))]$, $w_k = \hat{\gamma}_1 u_k (|\hat{\gamma}_1 u_k|^{\hat{\alpha}_1 - 1} - 1)\tan(\pi\hat{\alpha}_1 / 2)$, $u_k = 0.1 + 0.1k$, $k = 0,..,9$.

Step 5. The final estimates are the following: $(\alpha_1, \beta_1, \gamma_1, \delta_1) = (\hat{\alpha}_1, \hat{\beta}_1, \hat{\gamma}_1, \hat{\delta}_{11} - \hat{\gamma}_1 \hat{\beta}_1 \tan(\pi\hat{\alpha}_1 / 2)$.

Step 6. Compute the relative error as $err = (\alpha_0 - \alpha_1)^2 + (\beta_0 - \beta_1)^2 + (\gamma_0 - \gamma_1)^2 + (\delta_0 - \delta_1)^2$. If $err > error$ or number of iterations reaches *maxiter*, repeat step 1 to 5, renormalizing the sample with the new values $(\alpha_0, \beta_0, \gamma_0, \delta_0) \to (\alpha_1, \beta_1, \gamma_1, \delta_1)$.

The SAS macro *%estimate_stable (data=, var=, error=, maxiter=)* (see Appendix B) estimates the parameters of a stable distribution under the parameterisation S1, having the following arguments:

- data - the SAS dataset containing the values of the analysed variables;
- var – the random variable for which the parameters are estimated;
- error –t he maximum acceptable error for convergence;
- maxiter – the maximum number of iteration to convergence.

The SAS dataset *res* contains the final estimates for the parameters of the stable distribution.

The SAS macro was applied on simulated random variables $S(\alpha = 1.5, \beta = 0, \gamma = 1, \delta = 0;1)$, using a maximum error of 0.01 and a maximum number of iterations of 100; the estimators are unbiased and the relative error is negligible.

Table 1. Estimates of 100 simulated variables under $S(\alpha = 1.5, \beta = 0, \gamma = 1, \delta = 0;1)$

	Alpha	Beta	Gamma	Delta
Mean	1.502	0.003	1.004	0.002
Std. Dev.	0.060	0.112	0.044	0.208

3. Conclusions

Estimating the parameters of a stable distribution is a highly intensive task in terms of computational effort, yet the applications of the stable distributions justify this effort, especially in the stock markets, in order to have a proper calibration of risk indicators.

Although there are several software products dealing with the issue of simulating and estimating a stable distribution, SAS has no procedure for stable distributions. In this paper we proposed two macros for estimating the parameters of a stable distribution using McCulloch method and Kogon-Williams method; further developments are required for implementing a procedure for estimation of stable distribution using maximum likelihood method.

Appendix A. SAS macro for estimating the parameters of a stable distribution using McCulloch method

```
/* SAS datasets f and g contains tabulated values */;

data f;
input na nb a b;
cards;
2.439 0     2     0
2.439 0.1   2     2.16
2.439 0.2   2     1
2.439 0.3   2     1
2.439 0.5   2     1
2.439 0.7   2     1
2.439 1     2     1
2.5   0     1.916 0
2.5   0.1   1.924 1.592
2.5   0.2   1.924 3.39
2.5   0.3   1.924 1
2.5   0.5   1.924 1
2.5   0.7   1.924 1
2.5   1     1.924 1
2.6   0     1.808 0
2.6   0.1   1.813 0.759
2.6   0.2   1.829 1.8
2.6   0.3   1.829 1
2.6   0.5   1.829 1
2.6   0.7   1.829 1
2.6   1     1.829 1
2.7   0     1.729 0
2.7   0.1   1.73  0.482
2.7   0.2   1.737 1.048
2.7   0.3   1.745 1.694
2.7   0.5   1.745 1
2.7   0.7   1.745 1
2.7   1     1.745 1
2.8   0     1.664 0
2.8   0.1   1.663 0.36
2.8   0.2   1.663 0.76
2.8   0.3   1.668 1.232
2.8   0.5   1.676 2.229
2.8   0.7   1.676 1
2.8   1     1.676 1
3     0     1.563 0
3     0.1   1.56  0.253
3     0.2   1.553 0.518
3     0.3   1.548 0.823
3     0.5   1.547 1.575
3     0.7   1.547 1
3     1     1.547 1
3.2   0     1.484 0
3.2   0.1   1.48  0.203
3.2   0.2   1.471 0.41
3.2   0.3   1.46  0.632
3.2   0.5   1.448 1.244
3.2   0.7   1.438 1.906
3.2   1     1.438 1
3.5   0     1.391 0
3.5   0.1   1.386 0.165
3.5   0.2   1.378 0.332
3.5   0.3   1.364 0.499
3.5   0.5   1.337 0.943
3.5   0.7   1.318 1.56
3.5   1     1.318 1
4     0     1.279 0
4     0.1   1.273 0.136
```

```
4       0.2     1.266 0.271
4       0.3     1.25  0.404
4       0.5     1.21  0.689
4       0.7     1.184 1.23
4       1       1.15  2.195
5       0       1.128 0
5       0.1     1.121 0.109
5       0.2     1.114 0.216
5       0.3     1.101 0.323
5       0.5     1.067 0.539
5       0.7     1.027 0.827
5       1       0.973 1.917
6       0       1.029 0
6       0.1     1.021 0.096
6       0.2     1.014 0.19
6       0.3     1.004 0.284
6       0.5     0.974 0.472
6       0.7     0.935 0.693
6       1       0.874 1.759
8       0       0.896 0
8       0.1     0.892 0.082
8       0.2     0.887 0.163
8       0.3     0.883 0.243
8       0.5     0.855 0.412
8       0.7     0.823 0.601
8       1       0.769 1.596
10      0       0.818 0
10      0.1     0.812 0.074
10      0.2     0.806 0.174
10      0.3     0.801 0.22
10      0.5     0.78  0.377
10      0.7     0.756 0.546
10      1       0.691 1.482
15      0       0.698 0
15      0.1     0.695 0.064
15      0.2     0.692 0.128
15      0.3     0.689 0.191
15      0.5     0.676 0.33
15      0.7     0.656 0.478
15      1       0.595 1.362
25      0       0.593 0
25      0.1     0.59  0.056
25      0.2     0.588 0.112
25      0.3     0.586 0.167
25      0.5     0.579 0.285
25      0.7     0.563 0.428
25      1       0.513 1.274
;
run;

data g;
input a b g d ;
cards;
0.5     0       2.588 0
0.5     0.25    3.073 -0.061
0.5     0.5     4.534 -0.279
0.5     0.75    6.636 -0.659
0.5     1       9.144 -1.198
0.6     0       2.337 0
0.6     0.25    2.635 -0.078
0.6     0.5     3.542 -0.272
0.6     0.75    4.808 -0.581
0.6     1       6.247 -0.997
0.7     0       2.189 0
0.7     0.25    2.392 -0.089
0.7     0.5     3.004 -0.262
0.7     0.75    3.844 -0.52
0.7     1       4.775 -0.853
0.8     0       2.098 0
0.8     0.25    2.244 -0.096
0.8     0.5     2.676 -0.25
0.8     0.75    3.265 -0.469
0.8     1       3.912 -0.742
0.9     0       2.04  -0.099
0.9     0.25    2.149 -0.237
0.9     0.5     2.461 -0.424
0.9     0.75    2.886 -0.652
0.9     1       3.356 1
1       0       2     0
1       0.25    2.085 -0.098
1       0.5     2.311 -0.223
1       0.75    2.624 -0.383
1       1       2.973 -0.576
1.1     0       1.98  0
1.1     0.25    2.04  -0.095
1.1     0.5     2.205 -0.208
1.1     0.75    2.435 -0.346
1.1     1       2.696 -0.508
1.2     0       1.965 0
1.2     0.25    2.007 -0.09
1.2     0.5     2.125 -0.192
1.2     0.75    2.294 -0.31
1.2     1       2.491 -0.447
1.3     0       1.955 0
1.3     0.25    1.984 -0.084
1.3     0.5     2.067 -0.173
1.3     0.75    2.188 -0.276
1.3     1       2.333 -0.39
1.4     0       1.946 0
1.4     0.25    1.967 -0.075
1.4     0.5     2.022 -0.154
1.4     0.75    2.106 -0.241
1.4     1       2.211 -0.335
1.5     0       1.939 0
1.5     0.25    1.952 -0.066
1.5     0.5     1.988 -0.134
```

```
1.5     0.75    2.045 -0.206
1.5     1       2.116 -0.283
1.6     0       1.933 0
1.6     0.25    1.94  -0.056
1.6     0.5     1.962 -0.111
1.6     0.75    1.997 -0.17
1.6     1       2.043 -0.232
1.7     0       1.927 0
1.7     0.25    1.93  -0.043
1.7     0.5     1.943 -0.088
1.7     0.75    1.961 -0.132
1.7     1       1.987 -0.179
1.8     0       1.921 0
1.8     0.25    1.922 -0.03
1.8     0.5     1.927 -0.061
1.8     0.75    1.936 -0.092
1.8     1       1.947 -0.123
1.9     0       1.914 0
1.9     0.25    1.915 -0.017
1.9     0.5     1.916 -0.032
1.9     0.75    1.918 -0.049
1.9     1       1.921 -0.064
2       0       1.908 0
2       0.25    1.908 0
2       0.5     1.908 0
2       0.75    1.908 0
2       1       1.908 0
;
run;

%macro mc_culloch(data=,var=);

proc iml;
 use &data;
READ all var {&var} into x;
p = {0.05, 0.25, 0.50, 0.75, 0.95};
call qntl(q, x, p); /* compute 5th,
25th, 50th, 75th and 95th quantiles
*/;

 use f;
READ all var {na nb a b} into f;
 use g;
READ all var {a b g d} into g;
 n=nrow(g);
m=nrow(f);
na=(q[5]-q[1])/(q[4]-q[2]);
nb=(q[5]+q[1]-2*q[3])/(q[5]-q[1]);
pi=constant('pi');
*estimate alpha;
do i=1 to m-1 while(na>=2.439);
j=i+1;
if (f[i,1]<=na & na<f[j,1]) then
x1=f[i,1] ;
if (f[i,1]<=na & na<f[j,1]) then
x2=f[j,1] ;

if (f[i,2]<=abs(nb) &
abs(nb)<f[j,2]) then
y1=f[i,2] ;
if (f[i,2]<=abs(nb) &
abs(nb)<f[j,2]) then
y2=f[j,2] ;
end;
do i=1 to m;

if (f[i,1]=x1 & f[i,2]=y1) then
q11=f[i,3] ;
if (f[i,1]=x1 & f[i,2]=y2) then
q12=f[i,3] ;
if (f[i,1]=x2 & f[i,2]=y1) then
q21=f[i,3] ;
if (f[i,1]=x2 & f[i,2]=y2) then
q22=f[i,3] ;
end;

      if na<2.439 then alpha=2 ;
      else
      alpha=(q11*(x2-na)*(y2-
abs(nb))+
      q21*(na-x1)*(y2-
abs(nb))+q12*(x2-na)*(abs(nb)-y1)+
      q22*(na-x1)*(abs(nb)-
y1))/((x2-x1)*(y2-y1));

*estimate  beta;
do i=1 to m-1 ;
      j=i+1;
      if (f[i,1]<=na & na<f[j,1])
then
      x1=f[i,1] ;
      if (f[i,1]<=na & na<f[j,1])
then
      x2=f[j,1] ;
      if (f[i,2]<=abs(nb) &
abs(nb)<f[j,2]) then
      y1=f[i,2] ;
      if (f[i,2]<=abs(nb) &
abs(nb)<f[j,2]) then
      y2=f[j,2] ;
end;
do i=1 to m;
      if (f[i,1]=x1 & f[i,2]=y1)
then
      q11=f[i,4] ;
      if (f[i,1]=x1 & f[i,2]=y2)
```

```
then
      q12=f[i,4] ;
      if (f[i,1]=x2 & f[i,2]=y1)
then
      q21=f[i,4] ;
      if (f[i,1]=x2 & f[i,2]=y2)
then
      q22=f[i,4] ;
end;
      beta=sign(nb)*(q11*(x2-
na)*(y2-abs(nb))+
      q21*(na-x1)*(y2-
abs(nb))+q12*(x2-na)*(abs(nb)-y1)+
      q22*(na-x1)*(abs(nb)-
y1))/((x2-x1)*(y2-y1));

if abs(beta)>1 then
beta=sign(nb)*1;
      *estimate gamma;
do i=1 to n-1;
j=i+1;
if (g[i,1]<=alpha & alpha<g[j,1])
then
x1=g[i,1] ;
if (g[i,1]<=alpha & alpha<g[j,1])
then
x2=g[j,1] ;

if (g[i,2]<=abs(beta) &
abs(beta)<g[j,2]) then
y1=g[i,2] ;
if (g[i,2]<=abs(beta) &
abs(beta)<g[j,2]) then
y2=g[j,2] ;
end;
do i=1 to n;
if (g[i,1]=x1 & g[i,2]=y1) then
q11=g[i,3] ;
if (g[i,1]=x1 & g[i,2]=y2) then
q12=g[i,3] ;
if (g[i,1]=x2 & g[i,2]=y1) then
q21=g[i,3] ;
if (g[i,1]=x2 & g[i,2]=y2) then
q22=g[i,3] ;
end;

      f3=(q11*(x2-alpha)*(y2-
abs(beta))+
      q21*(alpha-x1)*(y2-
abs(beta))+q12*(x2-
alpha)*(abs(beta)-y1)+
      q22*(alpha-x1)*(abs(beta)-
y1))/((x2-x1)*(y2-y1));
      gamma=(q[4]-q[2])/f3;

      *estimate delta;
do i=1 to n-1;
j=i+1;
if (g[i,1]<=alpha & alpha<g[j,1])
then
x1=g[i,1] ;
if (g[i,1]<=alpha & alpha<g[j,1])
then
x2=g[j,1] ;

if (g[i,2]<=abs(beta) &
abs(beta)<g[j,2]) then
y1=g[i,2] ;
if (g[i,2]<=abs(beta) &
abs(beta)<g[j,2]) then
y2=g[j,2] ;
end;
do i=1 to n;
if (g[i,1]=x1 & g[i,2]=y1) then
q11=g[i,4] ;
if (g[i,1]=x1 & g[i,2]=y2) then
q12=g[i,4] ;
if (g[i,1]=x2 & g[i,2]=y1) then
q21=g[i,4] ;
if (g[i,1]=x2 & g[i,2]=y2) then
q22=g[i,4] ;
end;
      f4=(q11*(x2-alpha)*(y2-
abs(beta))+
      q21*(alpha-x1)*(y2-
abs(beta))+q12*(x2-
alpha)*(abs(beta)-y1)+
      q22*(alpha-x1)*(abs(beta)-
y1))/((x2-x1)*(y2-y1));
      csi=q[3]+gamma*sign(beta)*f4;
if alpha=1 then delta=csi;
else delta=csi-
beta*gamma*tan(pi*alpha/2);

      parms=j(1,4,0);
      parms[1,1]=alpha;
      parms[1,2]=beta;
      parms[1,3]=gamma;
      parms[1,4]=delta;
      create parms from
parms;append from parms;
quit;

/* dataset parms contains estimates
of stable distribution/;
```

```
data parms;set parms;
rename col1=alpha;
rename col2=beta;
rename col3=gamma;
rename col4=delta;
run;
```

Appendix B. SAS macro for estimating the parameters of a stable distribution using Kogon-Williams method

```
%include
"d:\proiecte\carte\mc_culloch.sas";

%macro    estimate_stable(    data=,
var=, error=, maxiter=);
%mc_culloch(data=&data, var=&var);

data _null_;
set parms;
call symput('delta0', delta);
call symput('gamma0', gamma);
call symput('alpha0', alpha);
call symput('beta0', beta);
run;

proc iml;
delta0=&delta0;
alpha0=&alpha0;
beta0=&beta0;
gamma0=&gamma0;
niter=1;
maxerror=1;
do   until   ((maxerror<&error)   |
niter=&maxiter);
if niter>0 then do;
use &data;
READ all var {&var} into x;
n=nrow(x);
x1=j(n,1,0);
re=j(10,1,0);
im=j(10,1,0);
w=j(10,1,0);
y=j(10,1,0);
z=j(10,1,0);
v=j(10,1,0);
q=j(10,1,0);
pi=constant('pi');
delta0 = delta0 ;
x1=(x-delta0)/gamma0;
do k=1 to 10;
u=0.1+0.1*(k-1);
rep=cos(u*x1);
imp=sin(u*x1);
w[k,1]=log(abs(u));
re[k,1]=mean(rep);
im[k,1]=mean(imp);
y[k,1]=log(-
0.5*log(re[k,1]**2+im[k,1]**2));
end;
x0t=j(1,10,1);
x=(x0t//w`)`;
b=inv(x`*x)*x`*y; alpha1=b[2,1];
gamma1=exp(b[1,1]/b[2,1]);
do k=1 to 10;
u=0.1+0.1*(k-1);
z[k,1]=atan(im[k,1]/re[k,1]);
v[k,1]=gamma1*u*(abs(gamma1*u)**(al
pha1-1)-1)*tan(pi*alpha1/2);
q[k,1]=u;
end;
x=(v`//q`)`;
c=inv(x`*x)*x`*z;
beta1=c[1,1];
gamma=gamma0*gamma1;
delta1=delta0+gamma0*c[2,1];
error=j(4,1,0);
error[1,1]=abs(alpha1-alpha0);
error[2,1]=abs(beta1-beta0);
error[3,1]=abs(gamma1-gamma0);
error[4,1]=abs(delta1-delta0);
maxerror=sum(error**2);
end;
alpha0=alpha1;
beta0=beta1;
gamma0=gamma1;
delta0=delta1;
niter=niter+1;
end;
if  alpha0  <=  0  then  alpha0=10**(-
10);
if alpha0>2 then alpha0=2;
if  gamma0  <=  0  then  gamma0=10**(-
10);
if beta0>1 then beta0=1;
if beta0<-1 then beta0=-1;
res=j(1,4,0);
```

```
delta0 = delta0 -
beta0*gamma0*tan(0.5*pi*alpha0);

res[1,1]=alpha0;
res[1,2]=beta0;
res[1,3]=gamma0;
res[1,4]=delta0;
create res from res;append from
res;
quit;

data res;set res;
rename col1=alpha;
rename col2=beta;
rename col3=gamma;
rename col4=delta;
run;
%mend;
```

References

Borak, S., Weron,R., 2010.STABLEREGKW: MATLAB function to estimate stable distribution parameters using the regression method of Kogon and Williams, Statistical Software Components, M429004, Boston College Department of Economics.

Härdle, W., Klinke, S., Turlach, B., 1995. XploRe - an interactive statistical computing environment. Springer Verlag, New York.

Kim, Y., Rachev, S., Bianchi, M., Mitov, I. And Fabozzi, F., 2010. Time Series Analysis For Financial Market Meltdowns. No 2, Working Paper Series In Economics, Karlsruhe Institute Of Technology (KIT), Department Of Economics And Business Engineering, http://Econpapers.Repec.Org/Repec:Zbw:Kitwps:2.

Kogon, S.M., Williams, D.B., 1998. Characteristic Function Based Estimation of Stable Distribution Parameters, in A Practical Guide to Heavy Tails: Statistical Techniques and Applications, R.J.Adler, R.E.Feldman, M.Taqqu eds., Birkhauser, Boston, 311-335.

Mandelbrot, B., 1963. The Variation Of Certain Speculative Prices. The Journal Of Business, Volume 36, Issue 4 (Oct., 1963). Pp. 394-419.

Nolan, J. P., 2011. Stable Distributions - Models For Heavy Tailed Data. Boston: Birkhauser. Unfinished Manuscript, Chapter 1 Online At Academic2.American.Edu/_Jpnolan.

Rachev, S. And Mitnik S., 2000. Stable Paretian Models In Finance. John Wiley, Series In Financial Economics And Quantitative Analysis, Chechester, New York.

Racheva-Iotova, Borjana., 2010. An Integrated System for Market Risk, Credit Risk and Portfolio Optimization Based on Heavy-Tailed Models and Downside Risk Measures. Dissertation, LMU München: Faculty of Mathematics, Computer Science and Statistics, see http://edoc.ub.uni-muenchen.de/12375/1/Racheva-Iotova_Borjana.pdf.

Rachev, S., 2003. Handbook of Heavy Tailed Distributions in Finance, Elsevier.

Rachev, S., Mitnik S., 2000. Stable Paretian Models In Finance, John Wiley, Series in Financial Economics and Quantitative Analysis, Chechester, New York.

Weron, R., 1996. On the Chambers-Mallows-Stuck method for simulating skewed stable random variables, Statistics and Probability Letters 28: 165{171. See also R. Weron, Correction to: On the Chambers-Mallows-Stuck method for simulating skewed stable random variables, Research Report HSC/96/1, Wroc law University of Technology, 1996, http://www.im.pwr.wroc.pl/~hugo/Publications.html.